

I - Identification de la fonction

Introduction :

.....

.....

Rappel : une donnée logique est constituée d'un seul bit et elle est mémorisable en utilisant une bascule.

Définition :

.....

.....

II - Les registres

Un registre est caractérisé par :

- *
- *
- *

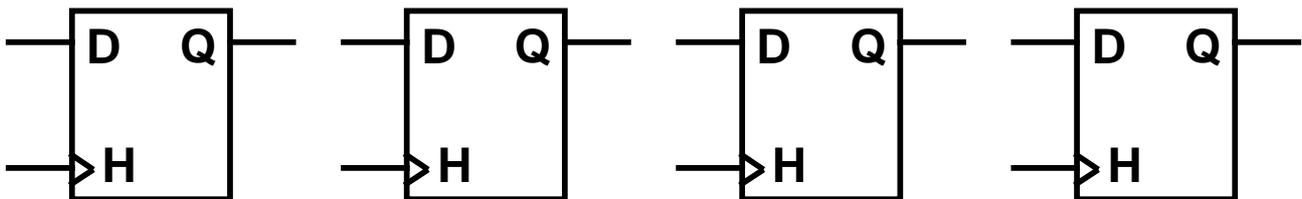
II - 1 - Le registre à entrée parallèle et à sortie parallèle

Ce registre est aussi appelé « *registre parallèle / parallèle* ».

.....

.....

Schéma d'un registre parallèle / parallèle 4 bits utilisant 4 bascules D :



Le chargement du registre s'effectue de manière globale à chaque front actif de l'horloge. Le mot binaire de 4 bits mémorisé est disponible à tout moment sur les sorties Q des bascules.

II - 2 - Le registre à entrée série et à sortie parallèle

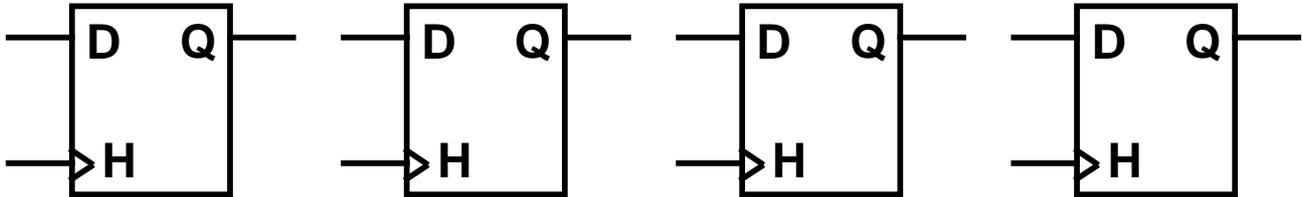
Ce registre est aussi appelé « *registre série / parallèle* ».

.....

.....

.....

Schéma d'un registre série / parallèle 4 bits utilisant 4 bascules D :



Remarques :

- * chaque sortie Q d'une bascule est reliée à l'entrée D de la bascule suivante
- * à chaque front actif de l'horloge les bits mémorisés dans le registre se décale d'un rang
- * un tel registre est appelé un **registre à décalage**
- * pour un registre à décalage à n bits il faudra attendre n impulsions sur l'horloge avant que le mot binaire soit entièrement mémorisé

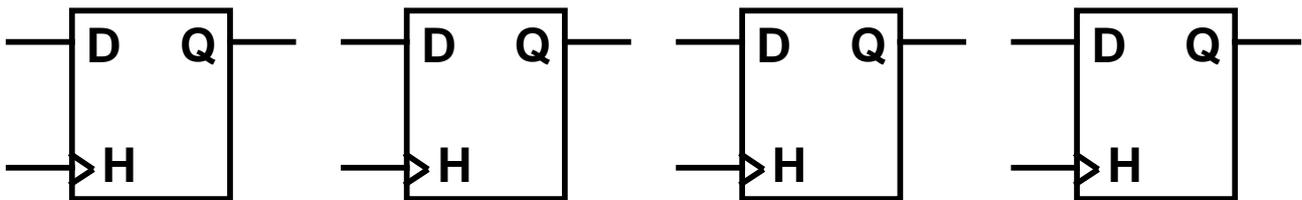
II - 3 - Le registre à entrée série et à sortie série

Ce registre est aussi appelé « *registre série / série* ».

.....

.....

Schéma d'un registre série / série 4 bits utilisant 4 bascules D :



Remarques :

- * le chargement du registre s'effectue bit à bit en 4 étapes, en envoyant 4 impulsions sur l'entrée d'horloge
- * la lecture du mot binaire mémorisé s'effectue également en 4 étapes, avec les 4 impulsions suivantes arrivant sur H
- * le premier bit entré dans le registre sera le premier bit sortie

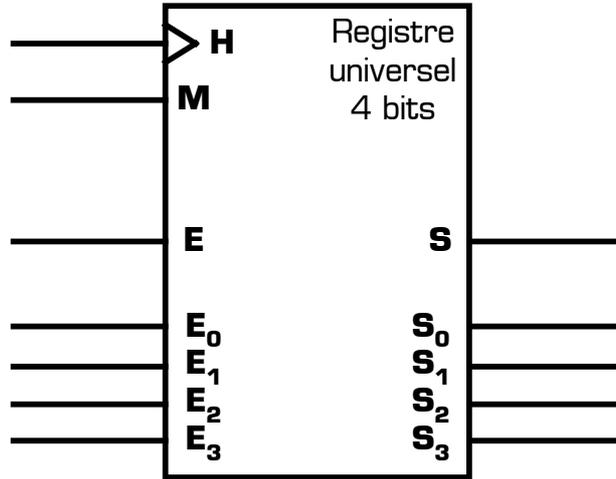
II - 4 - Le registre universel

.....

.....

- *
- *

Symbole d'un registre universel 4 bits :



L'entrée M permet de déterminer le mode de fonctionnement utilisé :

| M | Mode de fonctionnement | Remarques |
|---|------------------------|--|
| | | Les données sont lues bit à bit sur l'entrée E, et sont sorties du registre bit à bit sur la sortie S |
| | | Le registre est chargé d'un coup à partir des entrées E ₀ à E ₃ . La donnée mémorisée est disponible à tout moment sur les sorties S ₀ à S ₃ . |

III - Les mémoires

La somme d'informations que doit traiter tout système automatique nécessite l'utilisation de circuits ayant la capacité de conserver un grand nombre de données numériques. Ces circuits capables de recevoir et de restituer les informations portent le nom de « *mémoire* ».

III - 1 - Structure des circuits mémoire

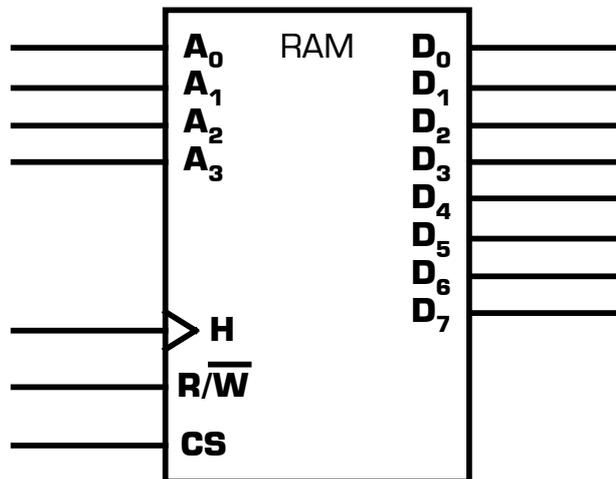
.....

.....

.....

.....

Symbole d'un circuit mémoire :



Circuit mémoire

Remarques :

Le bus d'adresse :

Le bus de données :

*

*

L'entrée R/W :

| R / W | Sens de circulation des données |
|-------|---------------------------------|
| | |
| | |

L'entrée CS :

| CS | Etat du boîtier mémoire |
|----|-------------------------|
| | |
| | |

L'entrée d'horloge H :

Dans l'exemple du circuit mémoire donnée en bas de la page 3, on remarque que :

- * le bus de données est composé de 8 bits : cela signifie que les mot binaire mémorisé on une taille de 8 bits, ce circuit mémorise donc des octets.
- * Le bus d'adresse est composé de 4 bits, il permet donc d'accéder à $2^4 = 16$ registre internes différents
- * La capacité de ce boîtier mémoire est donc de 16 octets

III - 2 - Caractéristiques des circuits mémoire

Un circuit mémoire est caractérisé par :

- * La taille des mots binaires qu'il peut mémoriser [exemple : 8 bits]
- * La quantité de mots binaires qu'il peut mémoriser [exemple : 16 ko]

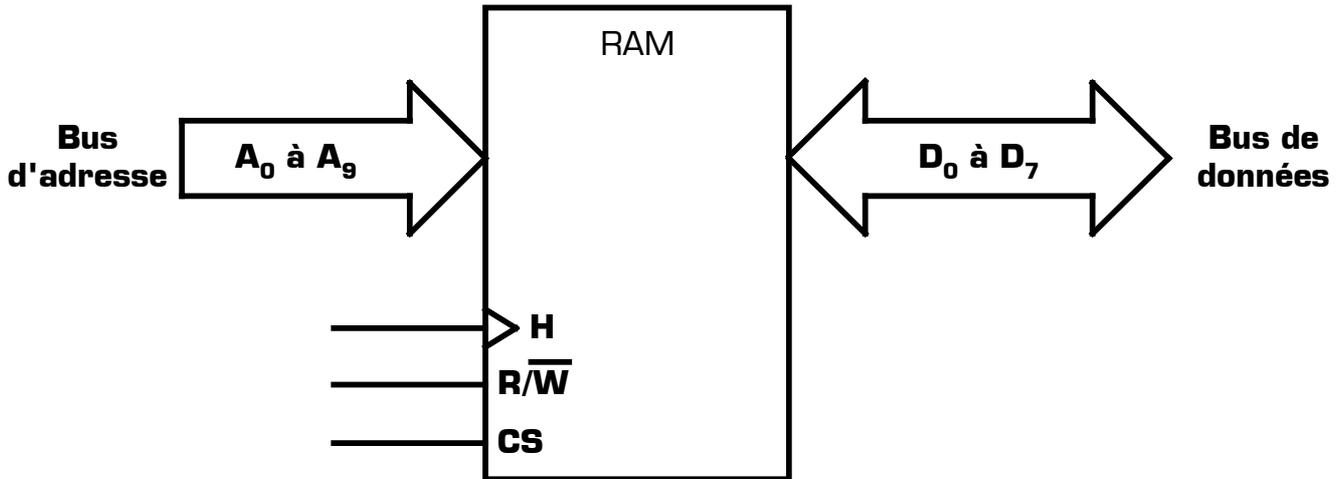
Rappel concernant les préfixes utilisés pour désigner les mémoires de grande capacité :

| Nom | Préfixe | Quantité équivalente |
|-----|---------|----------------------|
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |

De plus chaque circuit mémoire possède une ou plusieurs entrée(s) de validation du boîtier, noté parfois E (pour Enable = validation) parfois CS (pour Chip Select = sélection du boîtier). Ces entrées peuvent être active au niveau haut ou au niveau bas [voir le symbole du circuit ou la documentation constructeur pour le savoir].

III - 3 - Emploi de l'hexadécimal pour désigner les adresses mémoire

Prenons comme exemple le circuit mémoire suivant possédant un bus d'adresse de 10 bits :



Cette mémoire mémorise des octets car son bus de données est sur

Elle possède lignes d'adresse [A_0 à A_9] permettant d'adresser registres internes de 8 bits chacun. La capacité de ce boîtier mémoire est donc de

L'adresse du 1^{er} registre est :

L'adresse du 1024^{ème} registre est :

L'espace mémoire adressable par ce boîtier mémoire va donc de l'adresse à l'adresse, ce qui représente adresses différentes.

Avantage de l'hexadécimal pour représenter les adresses mémoire :

- * Avec très peu de chiffres l'hexadécimal permet de représenter les grands espaces mémoire, ce qui n'est pas le cas en binaire.
- * La conversion entre le binaire et l'hexadécimal est immédiate (il suffit de regrouper les bits par paquets de 4), ce qui n'est pas le cas du décimal.

Pour toutes ces raisons l'hexadécimal sera toujours le système de numération utilisé pour exprimer les adresses mémoires dans le plan mémoire d'un système numérique.

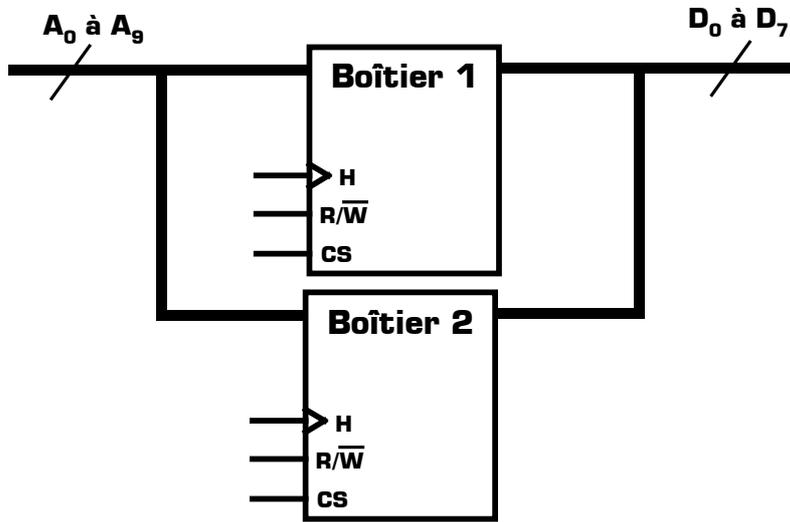
III - 4 - Utilisation de plusieurs boîtier mémoire pour augmenter la capacité

Le boîtier mémoire précédent a une capacité de 1 ko. Si on veut une mémoire de 2 ko il faudra alors utiliser 2 boîtiers différents :

.....

Il faut 11 lignes d'adresse pour adresser 2048 adresses différentes/ L'idée est alors la suivante : on va ajouter un 11^{ème} bit sur le bus d'adresse [noté A_{10}] qui va permettre de sélectionner un boîtier ou l'autre selon qu'il est à 0 ou à 1 :

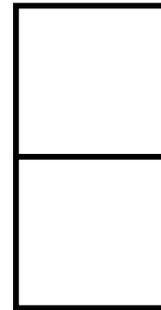
| A_{10} | Boîtier mémoire sélectionné |
|----------|-----------------------------|
| | |
| | |



Réalisation d'une mémoire de 2 ko avec deux boîtiers de 1 ko chacun

Conséquence :

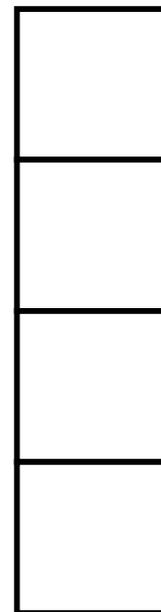
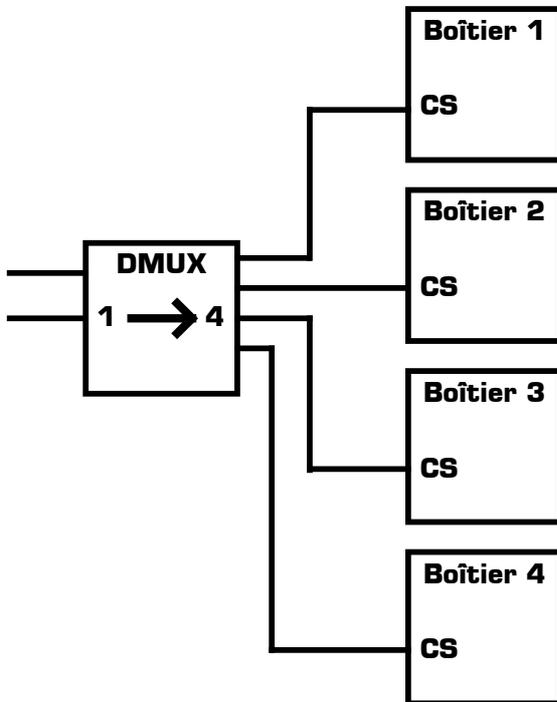
Les adresses $0000000000_{(2)}$ à $0111111111_{(2)}$ correspondent aux 1024 registres du boîtier 1 et les adresses $1000000000_{(2)}$ à $1111111111_{(2)}$ correspondent aux 1024 registres du boîtier 2. On peut alors représenter l'espace mémoire par le plan mémoire ci-contre. L'espace mémoire adressable va maintenant de l'adresse $000_{(16)}$ à l'adresse $7FF_{(16)}$, ce qui représente bien une capacité de 2 ko [2048 adresses].



Découpage du plan mémoire

Autre exemple : comment adresser 4ko en utilisant 4 boîtiers identiques de 1 ko chacun ?

La solution consiste à utiliser un démultiplexeur 1 vers 4, dont les 2 entrées d'adresse seront ajoutées au bus d'adresse :



Découpage du plan mémoire

Sélection des boîtier mémoire en fonction des bits de poids fort A_{10} et A_{11} du bus d'adresse :

| A11 | A10 | Boîtier sélectionné | Intervalle des adresse de chaque boîtier en binaire |
|-----|-----|---------------------|---|
| 0 | 0 | | |
| 0 | 1 | | |
| 1 | 0 | | |
| 1 | 1 | | |